

# Intellectual Ventures Technology Tutorial

April 22, 2022

**Dynamically adjust transmission parameters in response to changing line conditions.**

- U.S. Pat. Nos. 6,798,735, 7,817,532 and 8,369,275 (“’735 patent family”)

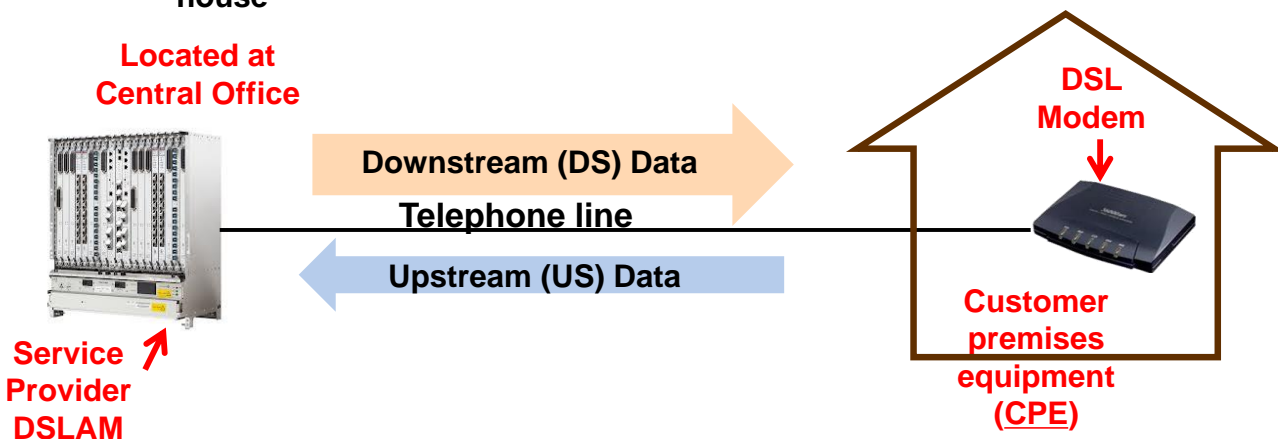
**Quickly initialize DSL transceivers so that they can begin to communicate data.**

- U.S. Pat. Nos. 6,647,068 and 7,272,171 (“’068 patent family”)

1A

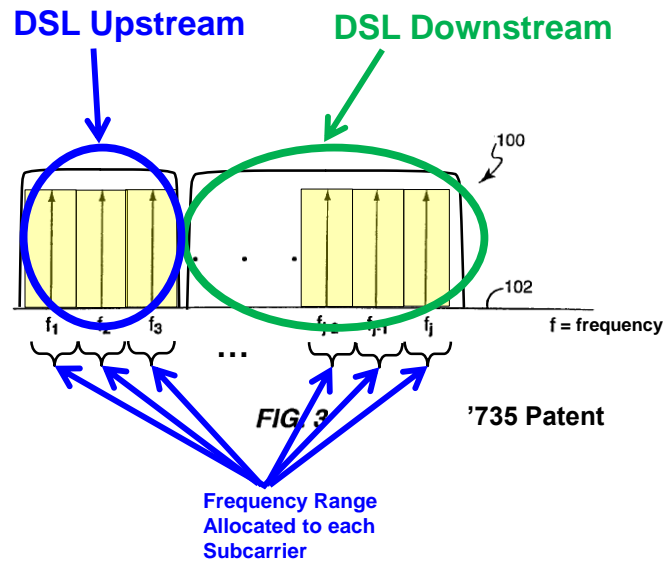
## DSL connections (e.g., over telephone lines) are bi-directional

- **Digital Subscriber Line (DSL) transmits data simultaneously in both directions**
  - Downstream: Service Provider (e.g., AT&T) to Customer
  - Upstream: Customer to Service Provider
- **Service Provider Equipment: DSL Access Multiplexer (DSLAM)**
- **Customer Premises Equipment: (CPE) DSL Modem at customer's house**



# DSL divides the frequency band into subcarriers

- DSL uses Discrete Multi-Tone Modulation (“DMT”)
- DMT divides the frequency band available on the telephone line into evenly spaced “subcarriers”
- Each subcarrier comprises a small portion of the entire frequency range



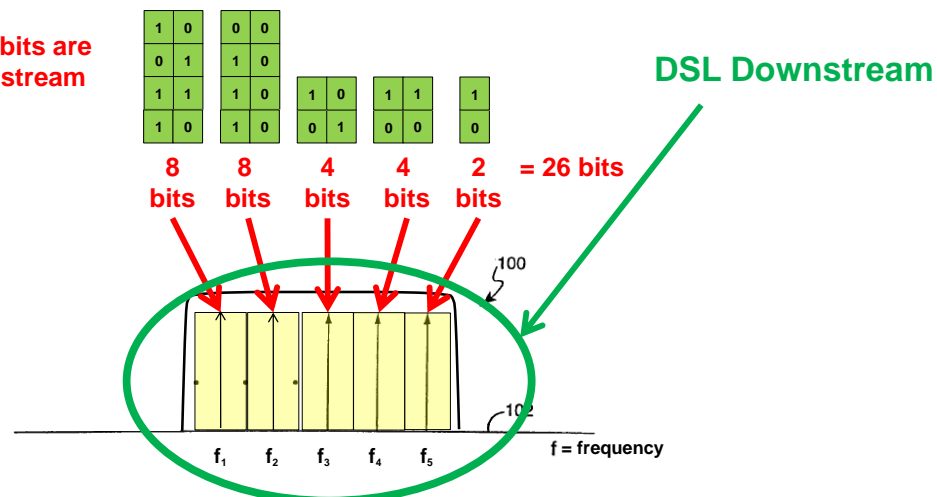
Intellectual Ventures Tutorial

2A

## Bits to be transmitted are allocated to subcarriers

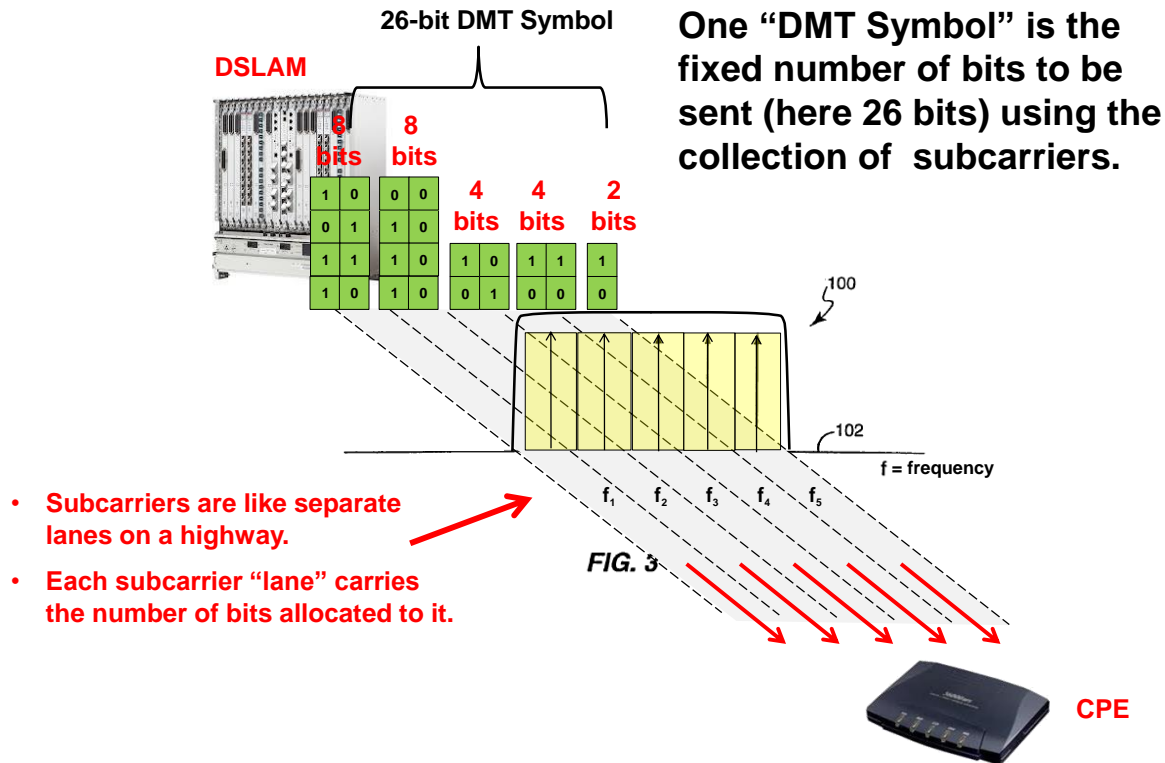
- Have digital signal comprising a given number of bits (representing information) to transmit – this example, 26 bits
- Divide signals into portions and allocate a number of bits to subcarriers (possibly zero bits to any particular subcarrier)

Example: A total of 26 bits are allocated to five downstream subcarriers



2B

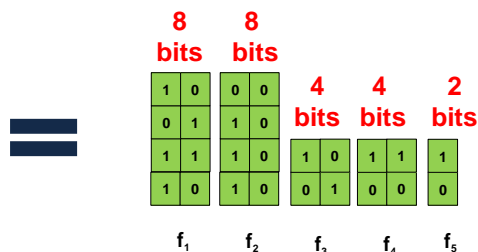
# Think of subcarriers as lanes on a highway



## Bit allocation for a DMT symbol

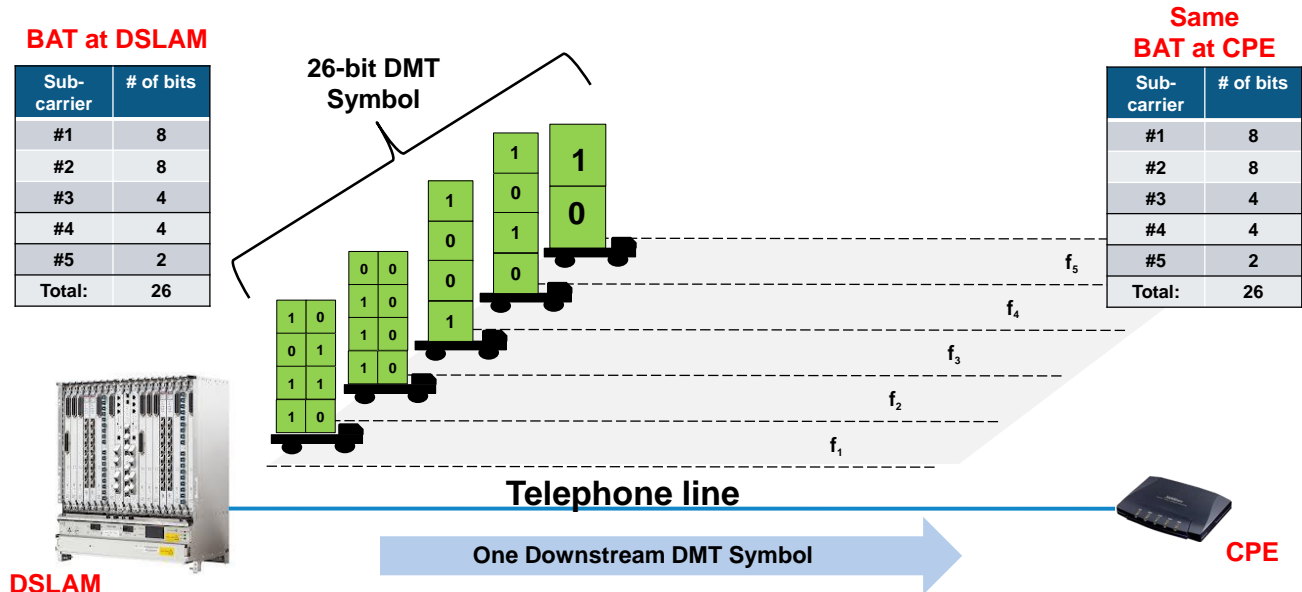
- A Bit Allocation Table (**BAT**) indicates the number of bits allocated to subcarriers
  - Example BAT for previous slide, with 26 Bits per DMT Symbol:

Subcarrier	# of bits
#1	8
#2	8
#3	4
#4	4
#5	2
Total:	26



# The transmitter and receiver must use the same BAT

- If the transmitter and receiver use different BATs for even one DMT symbol, the received signal may be decoded incorrectly and bit errors may occur.

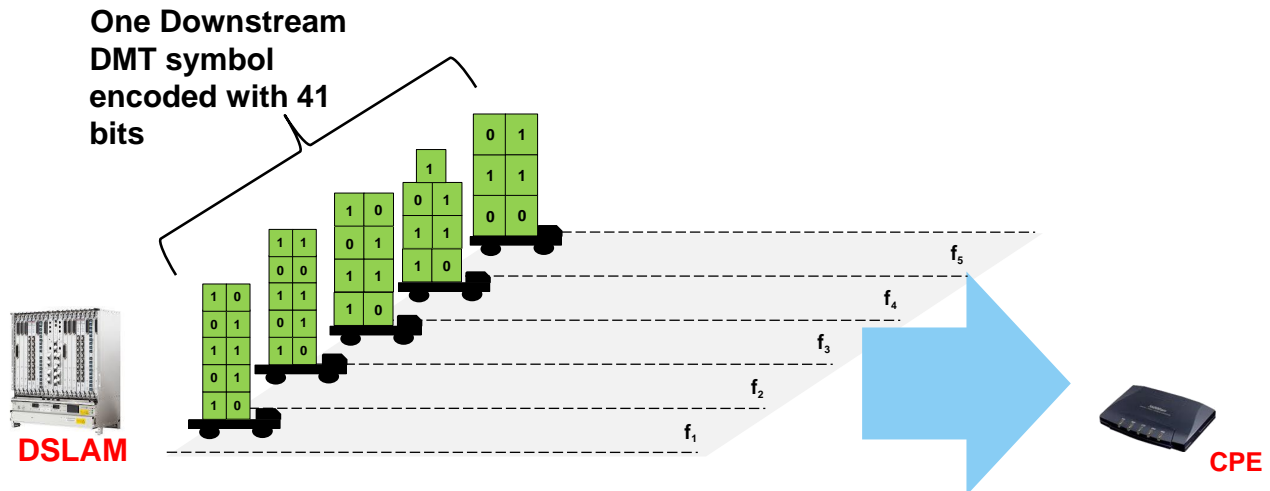


## '735 patent family – “adaptive bit allocation”

- Subcarriers encounter noise from various sources:
  - For example, electrical noise on the line and interference from other telephone lines
- If noise is too large relative to the received signal, “bit errors” may occur.
  - A sent “1” bit is interpreted as a “0” bit at the receiver, or a sent “0” bit is interpreted as a “1” bit at the receiver
- To avoid bit errors, a transmitter can transmit fewer bits on a subcarrier, resulting in a stronger signal per bit
- A subcarrier’s noise level can increase or decrease over time.
  - Therefore, a reliable procedure is needed to re-allocate bits to subcarriers (i.e., allow more bits or less bits) during data communication
- The '735 patent family provides a reliable procedure to re-allocate bits during communication**

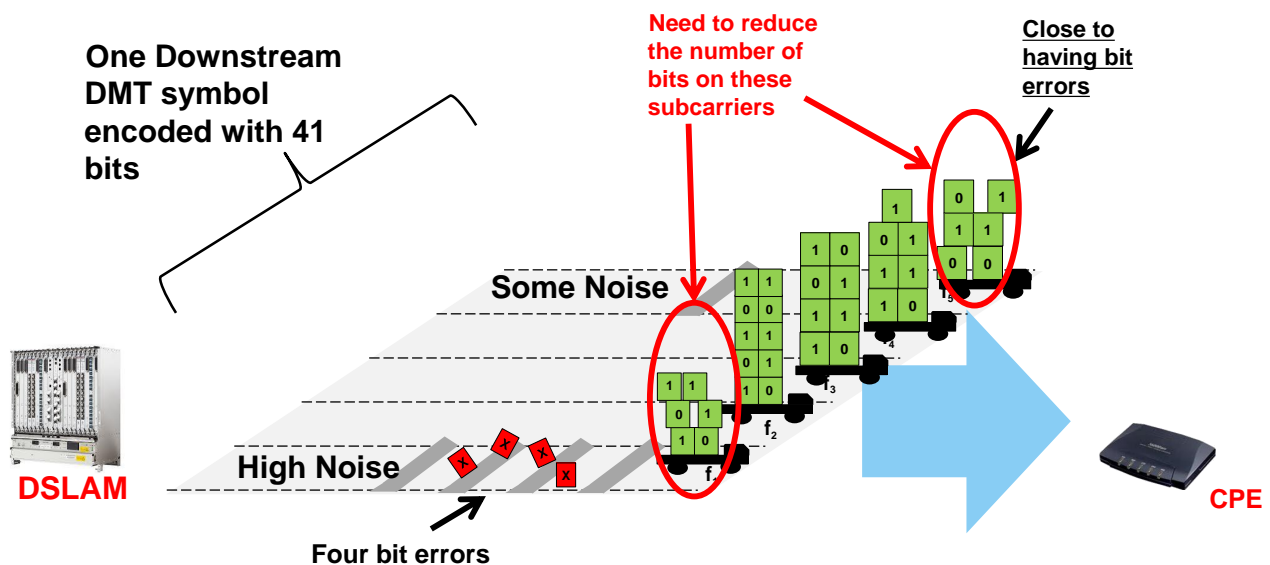
## With low noise, more bits per subcarrier

- A low noise level is like a smooth road: more and smaller blocks can be transported without having them fall off the truck
  - Low noise allows more bits to be sent using the same amount of transmitted power



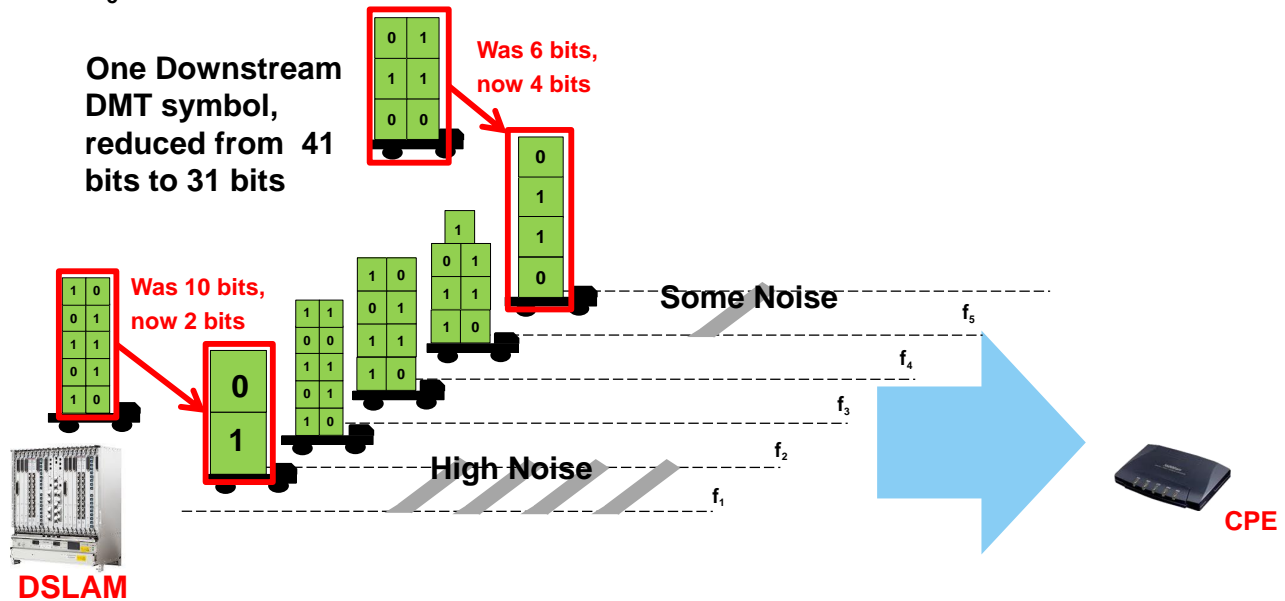
## Higher noise levels can cause bit errors

- Higher noise may result in bit errors
  - Noise is like bumps and debris in the road that make the blocks fall off their trucks
  - The higher the noise, the more likely it is that bits will be received in error on that subcarrier



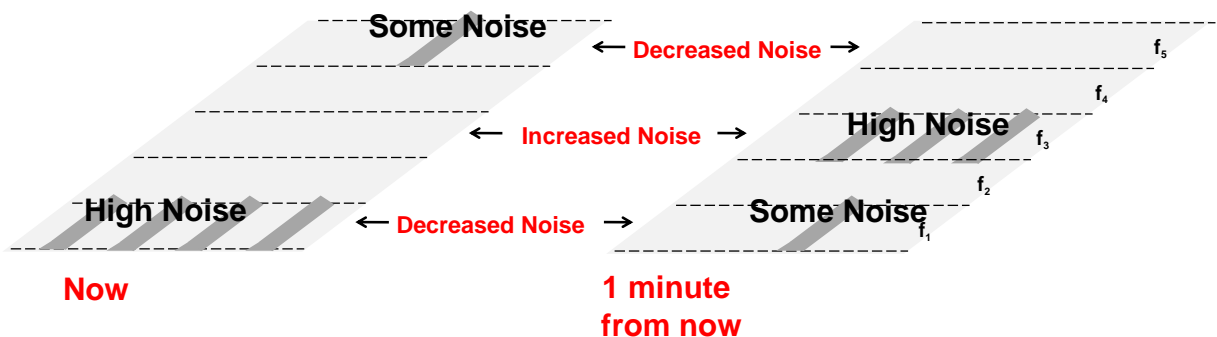
# Allocation of bits must be reduced on noisy subcarriers

- In the example below, the  $f_5$  subcarrier's allocation of bits is reduced from 6 to 4, and the  $f_1$  subcarrier's allocation of bits is reduced from 10 to 2
- With reduced allocation, no bit errors occur at the receiver, even on the  $f_1$  and  $f_5$  subcarriers



## Noise levels on subcarriers change over time

- The noise level on a subcarrier may increase or decrease over time
- The allocation of bits should be updated for those subcarriers with changed noise levels
  - Otherwise, if the noise level rises on a subcarrier, bit errors are more likely
  - On the other hand, if the noise level drops on a subcarrier, data rate capacity is being wasted

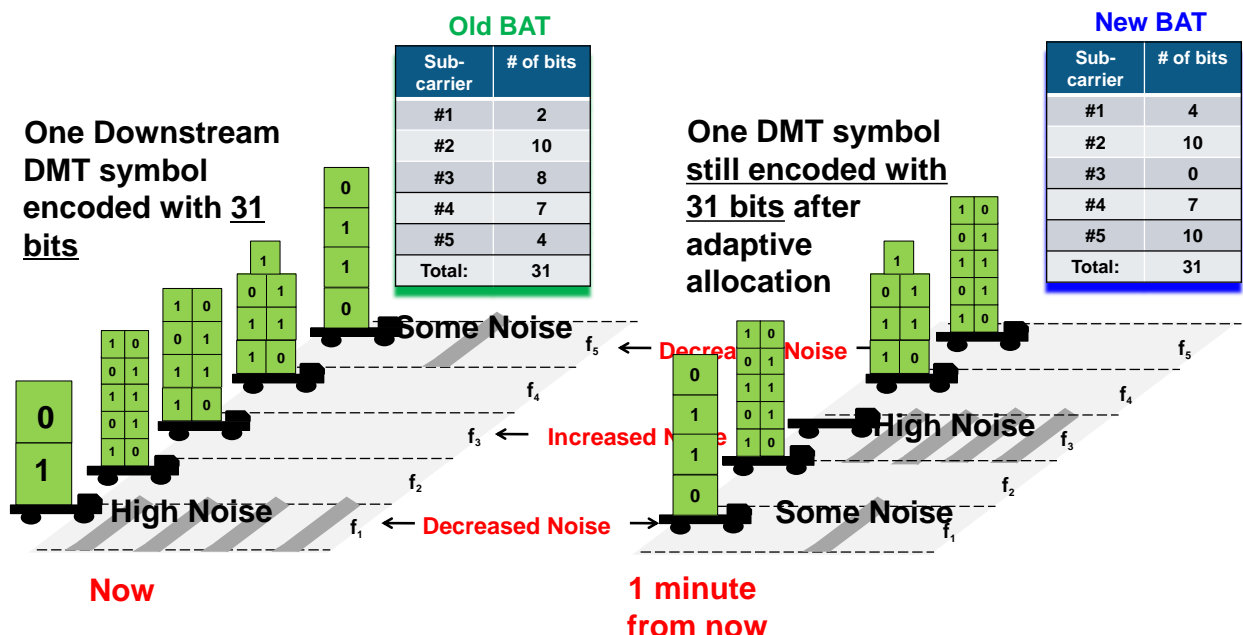


## “Adaptive allocation” can result in change of data rate or no change of data rate

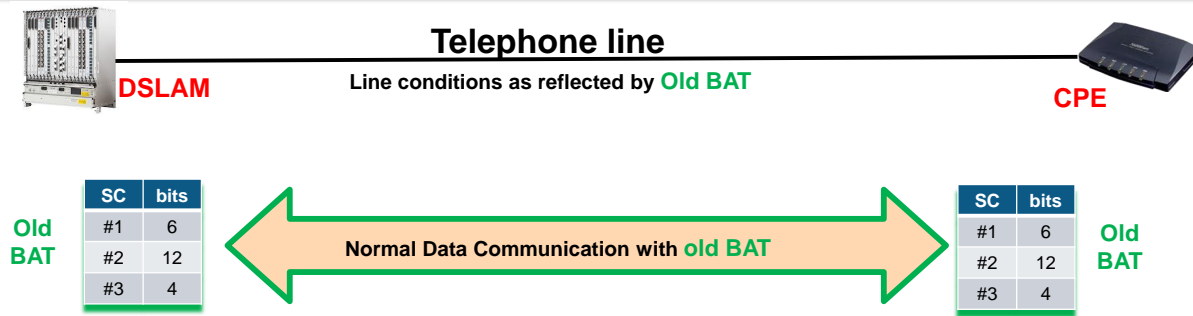
- In the previous example, the downstream DMT symbol is reduced from 41 total bits of information to 31 total bits. Because each DMT symbol carries less information, the data rate changes (decreases). This is known as “seamless rate adaptation” in the ADSL2/2+ and VDSL2 standards.
- However, if certain subchannels are allocated to carry more bits while others less, resulting in the total number of bits per DMT symbol remaining the same, there is no change of data rate. This is known as “bitswap” in the ADSL2/2+ and VDSL2 standards. (see next page)

## “Bit swap” adaptive allocation does not change the total number of bits

- Some subcarriers may improve while others degrade
- A “Bit Swap” reallocates bits without changing the total number of bits per DMT symbol



# Explanation of '735 patent family solution



- Initially, normal data communications using the old Bit Allocation Table (**old BAT**).
- But eventually, **line conditions change** – the '735 patent family provides a reliable procedure to adapt to changing line conditions. It does so by adjusting the BAT (to a **new BAT**) during ongoing communications.

# Explanation of '735 patent family solution

**TELEPHONE LINE  
CONDITIONS CHANGED**



**Step 1. A new Bit Allocation Table (new BAT) reflecting new line conditions is formed at CPE; '735 patent at col 5: line 35.**

SC	Bits	
#1	9	<b>New BAT</b>
#2	7	
#3	4	

**Step 2. The new BAT is communicated to the DSLAM; '735 patent at col 5: line 36.**

SC	Bits		SC	Bits	
#1	9	<b>New BAT (from CPE)</b>	#1	9	<b>New BAT</b>
#2	7		#2	7	
#3	4		#3	4	

**Step 3. When DSLAM is ready to use new BAT, DSLAM transmits a "FLAG" to the CPE. '735 patent at col 5: lines 36-39.**

**FLAG**

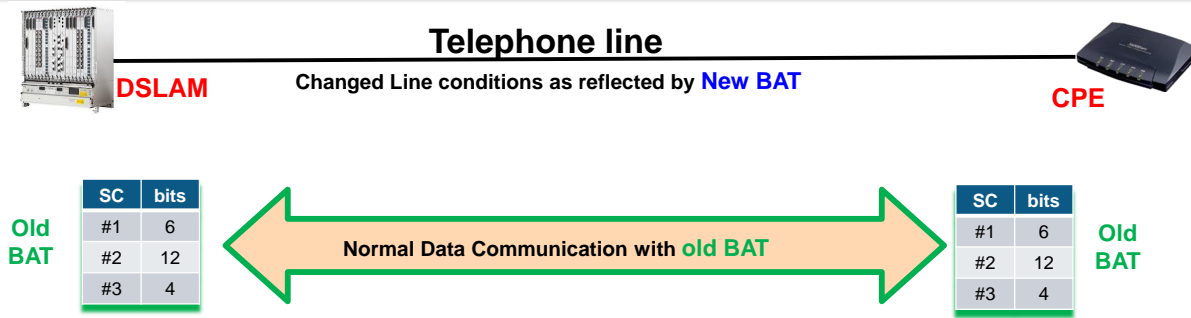
**Step 4. After a certain agreed upon number of symbols after FLAG, both DSLAM and CPE switch to using new BAT simultaneously; '735 patent at col 5: lines 39-40.**

SC	Bits		SC	Bits	
#1	9	<b>New BAT</b>	#1	9	<b>New BAT</b>
#2	7		#2	7	
#3	4		#3	4	

Normal Data Communication with **new BAT**



# Explanation of '735 patent family solution



- Initially, normal data communications using the old Bit Allocation Table (old BAT).
- But eventually, **line conditions change** – the '735 patent family provides a reliable procedure to adapt to changing line conditions. It does so by adjusting the BAT (to a **new BAT**) during ongoing communications.



## Asserted Patents

**Dynamically adjust transmission parameters in response to changing line conditions.**

- U.S. Pat. Nos. 6,798,735, 7,817,532 and 8,369,275 (“’735 patent family”)

**Quickly initialize DSL transceivers so that they can begin to communicate data**

- U.S. Pat. Nos. 6,647,068 and 7,272,171 (“’068 patent family”)

# What is Initialization?

- Initialization is a procedure to establish a link between two transceivers (e.g., DSLAM and CPE).
- When DSLAM and CPE first connect, they must:
  - Learn each other's capabilities and choose an operating mode.
  - Determine the noise and attenuation of each subchannel.
  - Determine the transmission "parameter values" that the data communication link will use, such as bit error rate and bit allocation table; '068 patent at col 4: lines 12-19.
- After initialization, transition to "normal" data communication, *i.e.*, online.



## When is Initialization Needed?

- Initialization is needed when, for example:
  - Power is interrupted
  - A DSLAM or CPE is reset
  - A DSLAM or CPE "wakes-up" (exits idle-state low power mode)
  - There is a significant change in line conditions that cannot be handled otherwise, e.g., by adaptive bit allocation - '735 family
- Problem: Full initialization takes a long time
  - Users don't want to wait ~30-60 seconds for full initialization before they can get online
  - line conditions may keep changing, causing repeated initializations



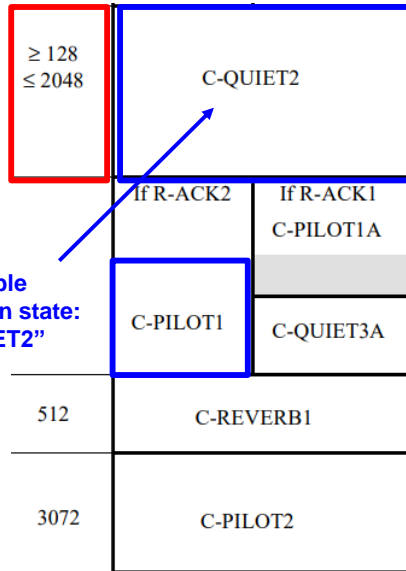
# Initialization process is made up of numerous "initialization states" each having a "state length"

Think of the initialization process as a baseball game and an "initialization state" as one inning of that game – state 1 (C-QUIET2) must complete before moving to state 2 (C-PILOT1) and so on.

State Length  
(number of  
DMT symbols)

Initialization  
States

Start of  
Initialization  
Process



Example  
initialization state:  
"C-QUIET2"

Time



Portion of Figure 10-5/G.992.1 – Timing diagram of the initialization sequence – Part 1



DSL  
CPE

DSLAM

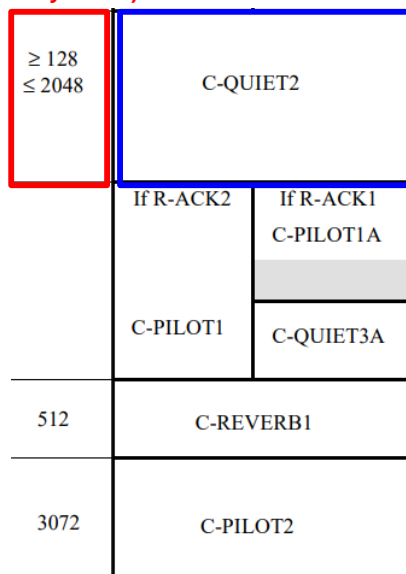
## Some states have fixed state length and others have "variable state length"

**Variable state length:**  
transceiver can wait  
between 128 and  
2048 DMT symbols,  
at its own discretion,  
before moving to the  
next state

State Length  
(number of  
DMT symbols)

Initialization  
States

Start of  
Initialization  
Process



Time



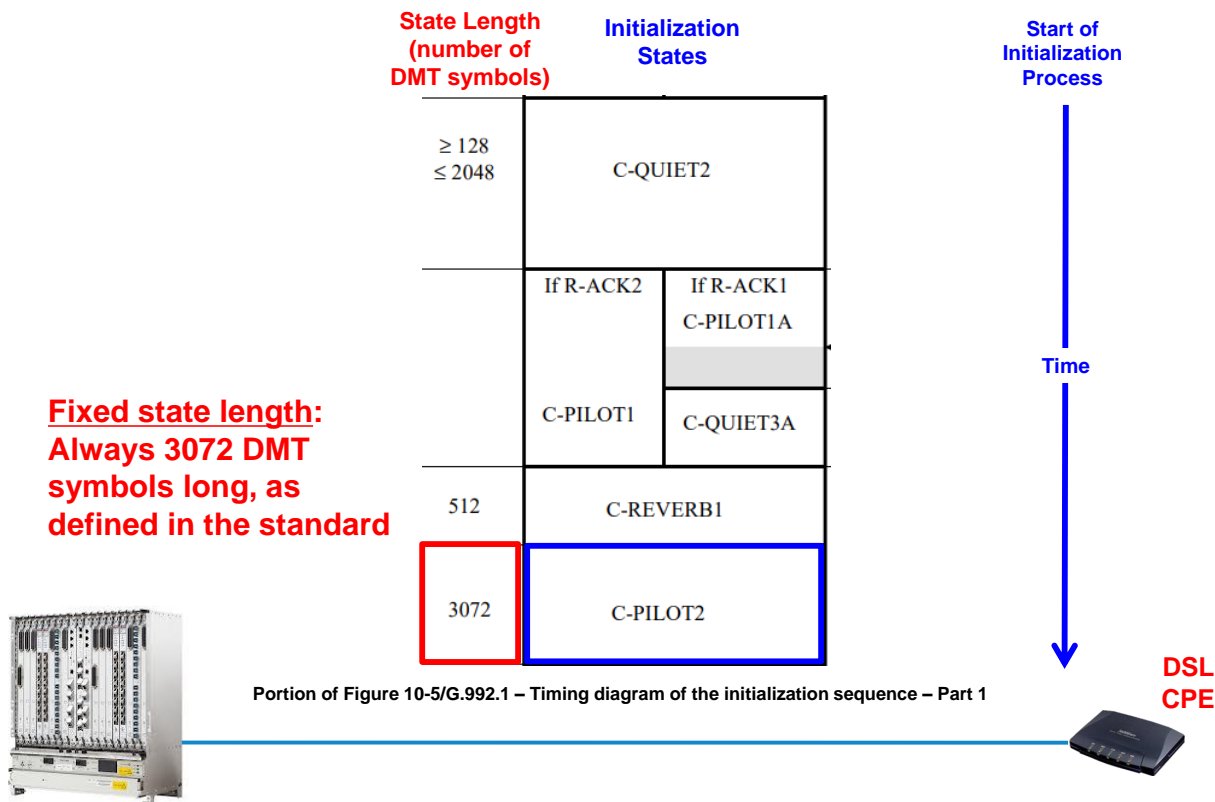
Portion of Figure 10-5/G.992.1 – Timing diagram of the initialization sequence – Part 1



DSL  
CPE

DSLAM

# Some states have fixed state length and others have “variable state length”



DSLAM

Intellectual Ventures Tutorial

12A

## Variable Length Initialization States

- What is a variable length initialization state?
  - An initialization state that does not have a fixed length (duration) specified in a standard
    - A standard may specify a lower limit (e.g., 128 symbols) and/or upper limit (e.g., 2048 symbols), but the exact length of the state is selected by one or both transceivers (i.e., DSLAM and CPE) during initialization
  - Useful because the time needed for one or both transceivers to complete internal processing steps may vary depending on the transceivers (i.e., make and model) and line conditions
  - Allows one transceiver (prior art) or both transceivers ('068 patent) to ensure they have time to perform necessary tasks, while minimizing the user's waiting time to get online.

# Problem and '068 Patent Family Solution

- Problem with prior methods: only one transceiver controls the length of the initialization states
  - Some variable length states may be too short for the transceiver that does not control the length (that other transceiver needs more time).
  - If the state lengths were controlled by both transceivers, that would enable a reduction in the initialization time.
- The '068 Patent Family Solution:
  - Both the transmitter and the receiver control the state length, resulting in faster and better initialization.

## Example of '068 patent family solution

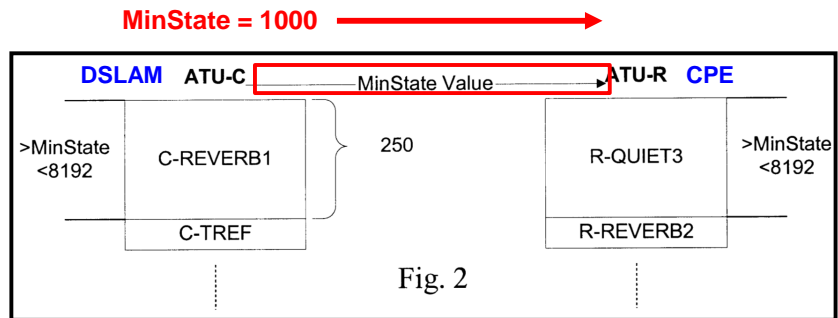
- '068 patent solution for variable state length:
  - Both transceivers control the length of the variable length initialization state.
  - For example, DSLAM sends to CPE a value indicating a minimum length it needs for the current variable length initialization state.
  - The CPE sends a signal to DSLAM to indicate the state's end, but only after waiting at least the indicated minimum length previously received from the DSLAM.
- Analogy
  1. I will give a speech, and we agree it will end when you yell "stop."
  2. Before the speech, I tell you "Give me at least one minute before yelling 'stop'"
  3. The speech starts. You have to wait at least one minute before yelling "stop."
  4. You yell "stop" and the speech ends.

In the prior methods, step (2) doesn't exist so I can't specify a minimum length for my speech. Thus, you might yell "stop" too early, cutting my speech short, or long after my speech has ended, wasting time.

## Both transceivers control length of variable state – as explained in the '068 patent

- Analogy
  - I will give a speech, and we agree it will end when you yell “stop.”
  - Before the speech, I tell you “Give me at least one minute before yelling ‘stop’”
  - The speech starts. You have to wait at least one minute before yelling “stop.”
  - You yell “stop” and the speech ends.

**DSLAM sends MinState value to CPE indicating the minimum length of the state. For example, 1000 DMT symbols. '068 patent at col 5: lines 13-18.**

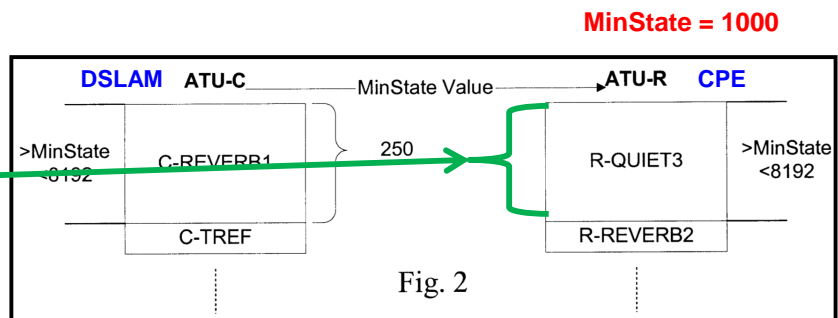


**Figure 2, '068 Patent**

## Both transceivers control length of variable state – as explained in the '068 patent

- Analogy
  - I will give a speech, and we agree it will end when you yell “stop.”
  - Before the speech, I tell you “Give me at least one minute before yelling ‘stop’”
  - The speech starts. You have to wait at least one minute before yelling “stop.”
  - You yell “stop” and the speech ends.

**The CPE must wait at least 1000 DMT symbols before the CPE could send R-REVERB2 to the DSLAM indicating end of state. '068 patent at col 5: lines 18-21.**

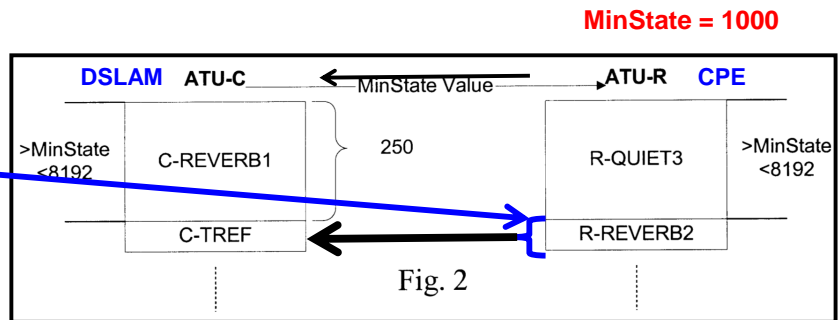


**Figure 2, '068 Patent**

Both transceivers control length of variable state – as explained in the '068 patent

- Analogy
  1. I will give a speech, and we agree it will end when you yell “stop.”
  2. Before the speech, I tell you “Give me at least one minute before yelling ‘stop’”
  3. The speech starts. You have to wait at least one minute before yelling “stop.”
  4. You yell “stop” and the speech ends.

**After at least 1000 DMT Symbols, the CPE decides when exactly to end the state, and sends R-REVERB2 to signal the end. '068 patent at col 5: lines 21-22.**



### Figure 2, '068 Patent